## **PATENT ABSTRACTS OF JAPAN**

(11) Publication number: 09305440 A

(43) Date of publication of application: 28 . 11 . 97

(51) Int. CI

G06F 11/30 G06F 11/00 G06F 13/00

(21) Application number: 08115984

(22) Date of filing: 10 . 05 . 96

(71) Applicant:

MATSUSHITA GRAPHIC COMMUN

SYST INC

(72) Inventor:

**SUZUKI TAKU** 

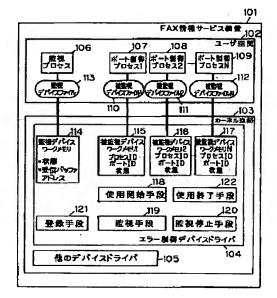
## (54) ERROR PROCESSING METHOD

## (57) Abstract:

PROBLEM TO BE SOLVED: To simplify an error process, to mitigate the load of the error process, and to evade a state wherein a device can not be controlled owing to the abnormal end of an operation management process in a processor consisting of plural operating on a multitask monitor and an operating system.

SOLUTION: This method has a monitor process which monitors an error and processes to be monitored; if an error occurs, information to be reported from a monitored process to the monitoring process is previously held before the error occurrence to the monitored process and reported to the monitoring process 106 after the error occurrence to the monitored process. If the abnormal end of the monitoring process 106 is detected during the error monitoring by the monitoring process 108, the system is reset and the monitoring process 106 is started again.

COPYRIGHT: (C)1997,JPO



THIS PAGE BLANK (USPTO)

(18)日本国格許庁 (JP)

糍 4 羋 华 噩 4 (12)

(11)特許出職公開每号 € 特開平9-305440

(43)公開日 平成9年(1997)11月28日

数形翻形

(51) Int CI.		40000000000000000000000000000000000000	庁内監理番号	F I			技権認
	1/30			G06F	11/30	×	
-	11/00	3.1.0			11/00	310C	
==	3/00	3 5 1			13/00	351M	

(全 12 員) **建型部次 未超次 部次項の数10 OL** 

(22) 出國日 平成8年(1996) 5月10日	(71) 出國人 000187738 東京都目編 東京都目編 (72) 発明者 時本 卓 東京都目編 電送株式会 (74) 代理人 弁理士 前	000187738 松下電送株式会社 東京都目風区下目開2丁目3番8号 傍木 東京都目風区下目開2丁目3番8号 東送株式会社内 弁理士 稿本 智之 (外1名)
	•	

1

(54) [発現の名様] エラー処理方法

ィングシステム上で動作する複数のプロセスからなる処 理装置において、エラー処理の簡潔化、エラー処理によ る負荷の軽減と運用管理プロセスの異常終了によって装 置が制御不能になる状況を回避することを目的としてい 【瞑題】 本発明は、マルチタスクモニタやオペレーテ

【解決手段】 エラー監視を行う監視プロセスと、複数 の被監視プロセスと、を有し、エラー発生時に前配被監 **視プロセスから前配監視プロセスに通知すべき情報を前** 配被監視プロセスでのエラー発生前に予め保持し、前記 プロセスに通知する、構成とした。また、監視プロセス 出された場合には、システムリセットを実行し再び監視 披監視プロセスでのエラー発生後に前記情報を前記監視 によるエラー監視実行中に監視プロセスの異常終了が検 プロセスを起動する構成とした。

6 FAX情報サービス装置 エラー製御デバイスドライバ 使用棒了手段 監視停止手段 -105 使用開始手段 監視手段 伯のデバイスドライバ 2 登録手段 2 東部でプ まださ

特許請求の範囲]

の被監視プロセスと、を有し、エラー発生時に前配被監 精束項1】 エラー監視を行う監視プロセスと、複数 見プロセスから前配監視プロセスに通知すべきエラー情 育報を前記監視プロセスに通知することを特徴とするエ し、前記被監視プロセスでのエラー発生後に前記エラ・ 段を前記被監視プロセスでのエラー発生前に予め保持

システム起動時に呼出されエラー情報を 保持する領域を蛰込み可能な状態にする使用開始手段を 有することを特徴とする請求項1配数のエラー処理方 請求項2】

ラー処理方法。

【請水項3】 エラー発生の際に前配被監視プロセスか ら前配監視プロセスに通知すべきエラー情報を保持する 処理を被監視プロセス起動時に実行する登録手段を有す ることを特徴とする請求項1配載のエラー処理方法。 1

情報を読み出して前配監視プロセスに通知する監視手 【請求項4】 監視プロセスにより起動された後命令受 **付け待ちの状態で待機し、エラー情報認識後にそのエラ** 段を有することを特徴とする請求項1記載のエラー処理

热下

【請求項5】 エラー発生による被監視プロセス終了時 【精水項6】 監視プロセスにより起動され監視手段に にエラー情報を監視手段に認識させる使用終了手段を有 対して監視停止命令を通知して監視を終了させる監視傳 **止手段を有することを特徴とする請求項4配載のエラー** することを特徴とする請求項4配載のエラー処理方法。 処理方法。

【請求項7】 監視プロセスが通常のイベント処理を実 を有して構成されることを特徴とする請求項5配載のエ 行するスレッドとエラー監視処理を実行するスレッドと ラー処理方法。

₫

ロセスと、を有し、この複数の被監視プロセスの異常終 了状態を各々検出し、前記被監視プロセスが異常終了し た場合に起動し異常終了した被監視プロセスを特定する 【開水項8】 エラー監視プロセスと、複数の被監視プ 骨報を前記エラー監視プロセスに通知するエラー処理を **東行することを特徴とするエラー処理方法。** 

プロセスを特定する情報を前記監視プロセスに通知する **ルと、を有し、前配いずれかのプロセスが異常終了しデ** 数の被監視プロセスと、これらのプロセスに対応して各 **を散けられプロセス実行阻始時にオープン状態となりプ** パイスファイルがクローズ状態となった場合にその異常 エラー監視を行う監視プロセスと、複 ロセス実行停止時にクローズ状態となるデバイスファイ ことを特徴とするエラー処理方法。 (間水項9)

のエラー処理方法。

[発明の詳細な説明 [0001] [発明の風する技術分野] 本発明は、マルチタスクのモ ニタやオペレーティングシステム上で動作する、複数の **処理単位 (タスクまたはプロセス) から構成される装置** こおけるエラー処理方法に関するものである。

[0002]

いる。この種の装置のエラー処理方法は、被エラー監視 【従来の技術】マルチタスク環境上で動作する複数の処 (プロセス) は、外部からのイベント受信を待ち、受信 灸にその処理を行うという動作をくり返すようになって 理単位から構成される装置においてては、各処理単位

プロセスがエラーを検知すると、エラー監視プロセスに、

**対して、プロセス間通信等の手段によりエラー通知が行** 

うようになっている。

【0003】以下、従来のエラー処理方法を具体的に図 9、図10を用いて説明する。図9はエラー監視プロセ スの処理フローを示し、図10は被エラー監視プロセス の処理フローを示している。

の受信待ちの状態で待機し(ステップ901)、イベン トが受信されると受信イベントが抜エラー監視プロセス からのエラー通知メッセージか否かをチェックし(ステ ップ902)、エラー通知であれば適切なエラー処理を **契行した後(ステップ903)、イベントの受信待ちの** [0004] 図9において、監視プロセスは、イベント 状態に復帰する (ステップ901)。

【0005】 政価イベントが通体のイベントであたば政 言イベントの処理を実行する (ステップ904)。

【0006】通常のイベント処理実行後、一定時間が極 過したか否かをチェックし (ステップ905)、一定時 問が経過していない場合は、ステップ901に復帰し、 **一定時間が経過していた場は、エラー監視となる全て** プロセス (V個) に対してその状態をチェックする以 の処理を実行する(ステップ906以下)。

ッセージを送信し (ステップ909) 、そのメッセージ 【0007】まず、カウンタ1を初期化し(ステップ9 カウンタ 1 をインクリメントし (ステップ901、ステ ップ908)、彼エラー監視プロセスに状態チェックメ 06)、カウンタ値がN以下であることを確認した後、

12)。また、ステップ909で正常なレスポンスを受 レスポンス待ちのタイムアウトが発生した場合には(ス テップ 911)、 適切なエラー処理を行う (ステップ 9 に対するレスポンス受信を待つ (ステップ910)。 そ **聞した場合もステップ910からステップ901ヘジャ** の後、エラーを報告するメッセージを受信した協合や、

まで繰り返し、全ての披エラー監視プロセスに対する処 [0008]以上一連の処理をカウンタ I がNに避する **組が終了した場合には、再びイベント受信待ちの状態に** 

を起動することを特徴とする請求項8又は請求項9記載

場合には、システムリセットを実行し再び監視プロセス

**こエラー制御手段が監視プロセスの異常終了を検出した** 

【請求項10】 監視プロセスによるエラー監視実行中

8

**梅屋片9-305440** 

復帰する (ステップ901)。

【0009】一方、エラー監視の対象である、被エラー監視プロセスの処理は、以下のとおりである。図10において、被エラー監視プロセスはイベントの受信待ちの状態で待機し(ステップ1001)、イベントが受信されると、そのイベントがエラー監視プロセスからの状態チェックメッセージか否かをチェックする(ステップ1002)。

【0010】発生したイベントが、エラー監視プロセス からの状態チェックメッセージでない場合には、通常のイベント処理を実行し(ステップ1003)、更に、その通常のイベント処理において障害が発生しているか否かをチェックし(ステップ1004)、正常であればイベント受信符もの状態に復帰する(ステップ100

1)。障害が検知された場合には、監視プロセスに対してエラー通知メッセージを送信し(ステップ100

5)、適切なエラー処理を実行した後(ステップ1006)、イベント受信符ちの状態に復帰する(ステップ1001)。

【0011】逆に、発生したイベントが、エラー監視プロセスからの状態デェックメッセージである場合には、 被エラー監視プロセスはその状態情報を監視プロセスに 送信する処理を行う(ステップ1007)

【0012】この様に、エラー監視プロセスが、定期的にエラー監視の対象となる全てのプロセスに対して状態チェックメッセージを送信し、これを受信した被エラー監視プロセスが状態情報を監視プロセスに送信するような監視形態ととる理由は、例えば、被エラー監視プロセスのプログラムの不具合等により、エラーの検出自体を行なえずプロセスが異常終了してしまい異常終了を監視プロセスに通知することができないという事態を回避するためである。

[0013]

(発明が解決しようとする課題)しかし、上述の従来技術の構成では、エラーを検出するために監視対象となるプロセスと監視プロセスとがメッセージを交換する必要があるため、本来の処理のほかにプロセス問通信処理的やタイマー処理を組み込まなければならず、ソフトウェアの処理が複雑になる。

【0014】また、監視プロセスが定期的に全ての監視対象プロセスに対して状態チェックメッセージを送信するため、特に監視対象数が大きい場合は野算資源が無駄に使用されることとなる。

【0015】また、装置管理機能を持つ監視プロセスが異常終了した場合には、他のプロセスが監視プロセスの異常終了を検知し、エラー処理を行わない限り、装置全体が制御不能になるという問題がある。この問題に対して、監視プロセスの障害検知をメッセージによる通知方式で行う場合には、各監視対象プロセスが逆に監視プロセスに対して状態チェックメッセージを送信することに

なり、ソフトウエアがますます複雑化し、負荷もさらに 大きくなることとなる。

【0016】本発明は、上述の課題に鑑みて為されたもので、マルチタヌクモニタやオペレーティングシステム上で動作する複数のプロセスからなる処理装置において、エラー処理の階級化、エラー処理による負荷の軽減と運用管理プロセヌの異常終了によって装置が制御不能になる状況を回避すること、を目的としている。

【課題を解決するための手段】上述の課題を解決するため、本発明は、エラー監視を行う監視プロセスと、複数の被監視プロセスと、複数の被監視プロセスと、を有し、エラー発生時に前記被監視プロセスから前記監視プロセスに通知すべき情報を前記被脱プロセスでのエラー発生前に予め保持し、前記被監視プロセスでのエラー発生後に前記情報を前記監視プロセスに通知する構成とした。また、監視プロセスによるエラー監視実行中に監視プロセスの異常終了が検出された場合には、システムリセットを実行し再び監視プロセスを起動する構成とした。

[0018] この構成により、監視プロセスが定期的に被監視プロセスの状態をチェックする必要がなくなり、エラー処理部のプログラム盘を飛躍的に減らすことが可能になり、また、エラー検知にプロセス問通信を使用しないので装置の計算資源の浪費を避けることができる。また、運用管理プロセスの異常終了によって装置が制御不能になる状況を回避することができる。

【発明の実施の形態】請求項1記載の発明は、エラー監視を行う監視プロセスと、複数の接監視プロセスと、を有し、エラー発生時に前記被監視プロセスから前記監視プロセスに通知すべきエラー情報を前記被監視プロセスでのエラー発生前に予め保持し、前記被監視プロセスでのエラー発生後に前記情報を前記監視プロセスに通知するものであり、エラー検出に必要な情報を被監視プロセスとは異なる場所に保持しておき事後的に監視プロセスに通知することにより被監視プロセスが異常終了した場合でも確実にエラー検出を行ない得る。

【0021】請求項3記載の発明は、請求項1において、エラー発生の際に前記接監視プロセスから前記監視プロセスに通知すべきエラー情報を保持する処理を接監視プロセス起動時に実行する登録手段を有するものであり、被監視プロセス起動の際にエラー通知に必要な情報の登録が実行されるため、被監視プロセス起動直後にそのプロセスのエラー監視が可能になる。

【0022】請求項4記載の発明は、請求項1におい

て、監視プロセスにより起動された後命令受付け符ちの状態で待機し、エラー情報認識後にそのエラー情報を読み出して前記監視プロセスに通知する監視手段を有するものであり、監視手段がエラー検出を開始、統行するため、監視手段を呼出すだけで全ての監視対象プロセスで生にたエラーを検知することが可能となる。

【0023】開来項5記載の発明は、請求項4において、エラー発生による被監視プロセス終了時にエラー情報を監視手段に認識させる使用終了手段を有するもので数を、監視手段を実行させることにより、エラー通知に必要な情報を被監視プロセス終了時に確実に監視プロセスに通知することが可能となる。

【0024】開来項6記載の発明は、開来項4において、監視プロセスにより起動され監視手段に対して監視停止命令を通知して監視を終了させる監視停止手段を有するものであり、監視手段によりエラー監視を開始し、監視終了手段によりエラー監視を中断でき、監視不要な処理に対してのエラー監視、エラー通知を避けることが可能となる。

1

【0025】 請求項7記載の発明は、請求項5において、監視プロセスが通常のイベント処理を実行するスレッドとエラー監視処理を実行するスレッドとを有して構成されるものであり、監視プロセス106がエラー監視をする処理としない処理を明確に区別することが可能になるため、エラー監視をしない処理において、不必要なエラー処理を実行する必要がなくなる。

【0026】請求項8記載の発明は、エラー監視プロセスと、複数の被監視プロセスと、を有し、この複数の被監視プロセスと、を有し、この複数の被監視プロセスの異常終了状態を各々検出し、前記被監視プロセスが異常終了した場合に起動し異常終了した被監視プロセスを特定する情報を前記エラー監視プロセスに通知するエラー処理を実行するものであり、被監視プロセスが異常終了した場合でも確実にエラー検出、エラー通知を行ない得る。

【0027】 請求項9配款の発明は、エラー監視を行う監視プロセスと、技数の被監視プロセスと、これらのプロセスに対応して各々数けられプロセス実行開始時にオープン状態となりプロセス実行停止時にクローズ状態となるデバイスファイルと、を有し、前配いずれかのプロセスが異常終了しデバイスファイルがクローズ状態となった場合にその異常プロセスを特定する情報を前記監視プロセスに通知するものであり、接監視プロセスの状態監視のデバイスファイルによりエラー検知を確実に行ない得る。

【0028】 請求項10記載の発明は、請求項8又は請求項9において、監視プロセスによるエラー監視実行申 末項9において、監視プロセスによるエラー監視実行しにエラー制御手段が監視プロセスの異常終了を検出した 場合には、システムリセットを実行し再び監視プロセス を起動するものであり、監視プロセスが監視中に異常終 で起動するものであり、監視プロセスが監視中に異常終 で起動するものであり、監視プロセスが監視中に異常終

と、から構成されている。更に、この基本のS空間10 は、後に詳述するように、複数のプロセスが実行される が不能になるという状況をなくすることが可能である。 ロセス (107~109) との双方が動作するようにな クシミリ通信ボードドライバ等との双方が動作し、ユー のデバイスドライバ105、倒えば、複数チャネルファ 3では、エラー制御デバイスドライバ104とそれ以外 ユーザ空間102と、 基本オペレーティングシステム あり、このFAX情報サービス装置101のメモリ空間 構成を示している。101はFAX情報サービス装置で イスを使用したFAX情報サービス装置のソフトウエア て具体的に説明する。図1は、本発明のエラー制御デバ ロセスを全て再生でき、監視プロセスがなく装置の操作 【0029】以下、本発明の実施の形態を図面を参照し ザ옆閏102では、騒視プロセス106とボート懸鐏プ (以下、OSという) が実行される基本OS空間103

【0030】さて、FAX情報サービス装配101は、電話回線と1対1に対応するN回のボート即御プロセス(107から109)を有し、複数のファクシミリ文軸を複数の情報ボックス(図示せず)に分類して蓄積している。ボート即御プロセス(107から109)はユーザから来る呼の婚債を持ち、婚厚後にユーザが所定の情報ボックスの情報の取出しの指示をした場合には、FBトーン等へ受信し、指定された情報ボックスに蓄積されたファクミリ文軸を送信する即御を行う。ユーザが情報ボックスへの難碌を指示した場合には、ボト即御プロセスはファクシミリ画像を受信し、指定の情報ボックスに踏鎖する即御を行う。

られた被監視デバイスファイル(110~112)にア り当て、各ワークメモリの状態フィーパドをクロースに イスファイルの状態を記述する領域を有している。FA 有し、これらのワークメモリは、それぞれ対応するデバ と、接觸視デバイスファイル(110~112)に対応 イル110に対応する監視デバイスワークメモリ114 本OS空間103に、作業領域として監視デバイスファ 06はFAX情報サービス装置の起動、停止処理を行う ポート囲御プロセス(107~109)に対応して設け 9) が動作中にエラー発生を検出した場合には、各々の はエラー制御デバイス104の初期化処理を呼び出し、 X情報サービス装置101が起動するときに、基本OS する被監視デバイスワークメモリ (115~117) を **쇱に、監視デバイスファイル110にアクセスしたボー** クセスしてエラー通知を行なう。一方、監視プロセス 1 【0031】このボート制御プロセス(107~10 **前記初期化処理はワークメモリ(114~117)を智** 【0032】エラー制御デバイスドライバ104は、基 ト慰御プロセスの実行監視を行うようになっている。

【0033】また、土ラー処理デバイスドライバ104

9

は、使用開始年段118と、監視年段119と、監視停止年段120と、登録年段121と位用終了年段122とを有しており、使用開始年段118は対応するデバイスフィイルがオープンされたときに基本のSにより呼び出され、使用終了年段122は対応するデバイスファイルがクローズされたときに基本のSにより呼び出されるプラムであり、また、監視手段119と監視停止手段120と登録年段121とは、基本のSが提供するシステムコール(以下では入出力制御命令と呼ぶ)により契行される固有のプログラムである。

【0034】以上のようなソフトウェア構成を採るFA X恰報サービス装置におけるエラー制御デバイスドライ ベによるエラー処理フローを図2乃至図8を用いて説明

[0035] 及初に使用网络手段118について説明する。使用网络手段118は、エラー制御デバイスドライバの一部を構成するプログラムであり、システムが立ち上がり、いずれかのプロセスが開始されてデバイスファイルがオーブンされた時に基本のSにより呼出され、そのプロセスに対応するデバイスファイルのワークメモリ(114~117)の状態フィールドをオーブン状態にするものである。図2は、その動作の詳細フローを示し

[0036] 監視プロセス106またはボート影響プロセス (107~109) が、それぞれ、職税デバイスファイル113または被職投デバイスファイル (110~112) をオープンしたときに、基本の5はオープンされたそのデバイスファイルをバラメータとして、コラー型御デバイスドライバ104の使用開始半段118を呼び出す。

(0037) 使用関始年段118は、パラメーケで指定されたデバイスフィルが監視デバイスフィルル113かどうかをチェックし (ステップ201)、監視デバイスフィル113がとうかをチェックし (ステップ201)、監視デバイスフィル113がオープンされているか否かを監視デバイスワークメモリ114の状態フィールドを参照することによりチェックする。その状態フィールドを参照することによりかエックする。その状態フィールドを参照することによりかエックする。その状態フィールドがオーブンの状態でカンしているということなので、ステップ203でエテーとリターンする。逆に、その状態フィールドの値がクローズであれば、ステップ204代様フィールドのゴボッローズであれば、ステップ204代様フィールドをオーブンに設定し、ステップ205にて成功をリターン

[0038] パラメータ指定のディイスファイルが監視 デバイスファイル113でなけわば、被監視ディイスファイルン12)であるので、ステップ206 において前記ファイルのデバイス番号をチェックする。 デバイス番号が不正であればエラーをリターンするが (ステップ208)、デバイス番号が正しければ、その デバイスファイルに対応するワークメキリ (115~1

17)の状態フィールドをチェックする (ステップ209)。その状態フィールド値がオープンであればエラーをリターンし (ステップ208)、クローズであればその状態フィールドをオープンに更難して (ステップ2100)、成功をリケーンする (ステップ211)。

【0039】以上一連の処理が使用関始寺段118の動作であり、これにより、監視デバイメファイル113及び実行される被監視デバイスファイル(110~110次行される被監視デバイスファイル(110~112)のワークメキリ(114~117)がオープン状態となり、エテー処理の監視プロセスの実行準確が魅う。

[0040]次に、監視手段119及び監視停止手段120について説明する。監視手段119及び監視停止手段120は、エラー前御デバイスドライバ104の一部を構成するプログラムであり、システム起動時に、監視プロセス106により基本OSを介して呼出され、その間被監視デバイスファイル(110~112)のエラー監視を実行するものである。

(0041) 図3は、監視手段119の動作の詳細フローを示している。監視プロセス106がオープンした監視ディイスファイル113に対して入出力制御コールを実行して監視手段119を呼び出すと、基本の5は、監視デバイスファイル113とその監視デバイスファイル113とその監視デバイスファイル113とをの監視デバイスファイルし13が指定した受信バッファアドレスをバラメータとして監視手段119を実行する。

[0042]まず、監視手段119は、バラメータで指定されたデバイスフィイルが監視デバイスファイル113かどうかをチェックし(ステップ301)、監視デバイスファイル113でなければエラーをリターンする(ステップ302)。監視デバイスファイル113であれば、監視デバイスワークメモリ114の状態フィールドを監視中に設定するとともに(ステップ303)、受信ベッファアドレスを指定アドレスに設定して受信符ち状態に入りエラー検出を待つ(ステップ304)。

【0043】この受信パッファに何等かの入力があれば待ち状態を終了し実行状態に入り、監視デバイスワークメモリ114の状態フィールドを再びオープン状態に設定した後(ステップ305)、受信パッファの内容を参照し監視停止手段が実行されたかどうかを判断する(ステップ306)。受信パッファに後述の監視停止手段120からの監視停止命令が設定されている場合は、監視停止をリターンし(ステップ307)、受信パッファに監視停止命令ではなくエラー情報が設定されていれば成功をリターンする(ステップ308)。このエラー検出後の処理は監視プロセス106が行なう。

なの必当は医児ノロで入してのかけより。 【0044】図4は監視停止手段122の動作の詳細フローを示している。監視プロセス106がオープンした監視デバイスファイル113に対して、監視停止手段120を指定した入出力制御コールを実行すると、基本の5は監視デバイスファイル113をパラメータとして監視停止手段120な実行する。監視停止手段120は、

バラメーダで指定されたデバイスファイルが監視デバイスファイル113でかりで 401)、監視デバイスファイル113でなければエラーが理とし (ステップ 402)、監視デバイスファイル 113であれば監視デバイスワーグ・モリ114の状態 フィールド値が監視中であるかチェックする (ステップ 403)。状態フィールド値が監視中でなければエラー 202とし (ステップ 402)、監視中でなければエラー 403)。状態フィールド値が監視中でなければエラー 202とし (ステップ 402)、監視中であれば監視デバイスワーグ・キリ114のバッファドレスフィールド で指定された受信バッファドレスフィールドで指定された受信バッファドレスフィールドで指定された受信バッファドレスフィールドで指定された受信バッファドレスフィールドで指定された受信バッファに監視停止をライトし (ステップ 405)、受信符与の状態になっている監視手段119を待ち状態から監視 停止状態に変更し、成功をリターンする (ステップ30 [0045]以上一連の処理が監視手段119及び監視 存止手段1200動作であり、エラー監視が必要な処理 に対して監視手段119の起動をかけるだけで、エラー が検出されるか監視停止処理が実行されるまでエラー監 視が検行されることとなる。また、監視停止処理も監視 手段119とは独立のプログラムである監視停止手段120により実行される。

1

図6はその動作の詳細フローを示している。

【0046】次に、登録手段121について説明する。登録手段121は、エラー樹御デバイスドライベ104の一部を構成し、エラー監視の対象となるボート制御プロセス(107万至109)の超動時に、各プロセスを特定するための情報を基本のS空間上のワークメモリに登録しておくためのプログラムであり、図5は、その動作の詳細フローを示している。

[0047] ポート制御プロセス (107万至109) がオープンされた被削御デバイスファイル (110~1 り2) に対して入出力制御コールを実行し、程峰手段を 呼び出すと、基本のSは、被削御デバイスファイル (1 10~112) とポート制御プロセス (107万至10 9) が提供するプロセス 1Dとポート番号とを含む情報 を受信する受信パッファのアドレスをパラメータとして 程録手段121を呼び出す。程録手段121は、パラメータのデバイスファイルが被監視ディイスファイル (1 10~112) か否かをチェックし (ステップ50

、 被監復デバイスファイル (110~112) でない場合はエラー処理をし (ステップ502)、 被監視デバイスファイル (110~112) である場合は、そのデバイスファイルの番号をチェックする (ステップ503)。 デバイスファイル番号が正であればエラー処理をし (ステップ502)、 デバイスファイル番号が正ければ、 前記受信パッファの内容を対応する故監視デバイスワークメモリ (115乃至117のメモリのうち1つ)にコピーして成功をリターンする (ステップ504、マスップ505)。

[0048]以上一連の処理が登録手段121の動作で

あり、エラーが発生するプロセスを特定する情報を基本 OSのワークエリア上に予め記述しておくことにより、 監視プロセスがエラー発生時にその情報を受け取り、エ ラー個所を認識できることになる。 [0049]次に、使用終了手段122について説明する。使用終了手段121は、エラー側御デバイスドライバ104の一部を構成するプログラムであり、いずれかのプロセスが終了しデバイスファイルがクローズした時に基本OSにより呼出され、そのプロセスに対応するデバイスファイルのワーグメモリ(114~117)の状態フィールドをクローズ状態にして、エラー発生プロセスを特定する情報を監視手段119の受傷バッファにコピーすることにより、スリーブ状態にあった監視手段119を実行状態にするものである。既に説明したように、この監視手段119によりエラー検出が行なわれる。

[0050] 監投プロセス106またはボート即御プロセス (107~109) が、それぞれ、オープン中の監 短デバイスファイル110または按監投デバイスファイル (110~112) をクローズしたときに、基本OS はそのデバイスファイルをバラメータとして、エラー超 匈デバイスドライバ104の使用棒了手段122を呼び出す。使用棒了手段122は、クローズされたデバイスファイルが被監視デバイスファイル113かどうかをチェックする (ステップ601)。

【0051】クローズされたデバイメファイルが、被照照ディイスファイル(110~112)でない(監視デバイスファイル(110~112)でない(監視デバイスフークメバイスファイルでもあり、独合に、照視デバイスロークメモリ114の状態フィードルが監視中であるか否かをチェック(ステップ610)、監視中ではければ成りをリターンする(ステップ611)一方、監視中であれば様本OSのリグートインターフェースによりFAX位にサービス装置101をリグートする(ステップ61

「0052】このシステムリプート (ステップ612) が実行されるのは、監視中の監視プロセス106が何ら かの障害により異常終了し、その原基本OSによってオ ープン状態であった監視デバイスフォイル113がクローズされる場合である。このように、監視中の監視プロ セス106が異常終了したときに、エラー前海デバイス ドライバ104が装置をリセットする機能を設け、装置 リブートの隔に監視プロセスを自動的に立ち上げるよう に設定することにより、監視プロセスが異常終了した際の エラー処理を別途設けることなく、システムが倒御不能 な状態で放置されるという状況を防止することが可能と 【0053】女に、クローズされたデバイスファイルが安監視デバイスファイル(110~112)である場合

2)\_に対応する被監視デバイスワークメモリ (115か ローズされた接照視デバイスファイル (110~11 は、監視デバイスワークメモリ114の受信パッファア クする(ステップ605)。 監視デバイスワークメモリ メモリ 1\_1 4の状態フィールドが監視中か否かをチェッ **ーズに設定し(ステップ604)、監視デバイスワーク** スファイルに対応する被監視デバイスワークメモリ (1 3) 、デバイス番号が正しければクローズされたデバイ が不正であればエラー処理を行ない (ステップ60 号のチェックを行い(ステップ602)、デバイス番号 について説明する。使用終了手段122は、デバイス番 ドレスフィールドで指定されている受信べッファに、ク をリターンし(ステップ608)、 監視中である場合 15から117のうちの10)の状態フィールドをクロ 1-14の状態フィールドが、監視中でない場合は、成功

ば、プロセスIDとポート番号) をコピーした後 (ステ ら117のうちの1つ) に格納されている情報 (例え 変更し、成功をリターンする(ステップ608)。 になっている監視手段119を待ち状態から実行状態に EUPをコールし(ステップ607)、受信待ちの状態 ップ606)、構本OSインターフェースであるWAK

1

等で異常終了し、基本OSがオープン状態にある前記ラ スファイルをクローズする場合と、当該プロセスが例外 セス(107~109)がエラーを検知して前記デバイ 作であるが、このように、被監視デバイスファイル (1 により被監視デバイスワークメモリ (115~117) が異常終了した場合にあっても、既に説明した登録手段 に、監視対象のポート制御プロセス(107~109) 場合には通常のエラー処理が実行されるが、後者のよう 10~112) がクローズされるのは、ボート毎海ブロ バイスファイルをクローズする場合と、がある。前右の ご通知することが可能となる。 こ予め格納してある情報が監視デバイスファイル113 【0054】以上一連の処理が使用終了手段122の敷 .必ず登録されるため、障害情報を確実に監視プロセン

る。また、エラー検知にプロセス間通信を使用しないの 理部のプログラム量を飛躍的に減らすことが可能にな ロセスの状態をチェックする必要がなくなり、エラー処 て装置の計算資源の浪費を避けることができる。 【0055】従って、監視プロセスが定期的に被監視フ

スの再起動等を実行するようになっている。 セスが異常終了した場合、エラー処理として前記プロセ サープス被閏101の短剰、存止、ボート艶御少ロセン 的に説明する。この監視プロセス106は、FAX情報 視、エラー処理を実行するフローを、図7を用いて具体 6がポート患匈プロセス (107~109) の状態階 一制御デバイスドライバを使用して、監視プロセス10 (107~109) の状態監視を行い、ポート制御プロ 【0056】以上のようなソフトウエア構成をもつエラ

【0057】まず、システムの起動処理が実行されると

制御コールを発行して監視停止手段により監視処理を停 合にはイベントの処理(通常の処理)を実行して(ステ 状態で (ステップ712)、イベントを受信すると、そ 以下の監視プロセス106本来の処理、つまり、システ 7717)。 ントである場合には、、監視停止手段を指定した入出力 の受信イベントがシステム停止か否かをチェックし(ス 4.停止イベントの処理を実行する。イベント受信待ちの 2)、エラー監視用スレッドを生成する(ステップ 70 09) を必要数 (電話回線数) 起動するとともに、監視 止し(ステップ715)、システム停止処理を実行し ップ714)、再UVイベント受信待ちの状態に復帰する テップ713)、それがシステム停止イベントでない場 3)。 生成されたメインスレッドでは、ステップ112 デバイスファイル113をオープンして(ステップ70 (ステップ116) 、監視プロセスを停止する (ステッ (ステップ112)。 受信イベントがシステム停止イベ (ステップ701)、ボート制御プロセス (107~)

段を指定した入出力制御コールを発行して監視手段11 受け取るバッファアドレスをパラメータとして、監視手 実行状態に遷移し、ステップ705から処理が再開され を呼び出した場合には、監視手段119は待ち状態から 了した場合と監視プロセス106が監視停止手段120 符つ。ポート制御プロセス(101~109)が異純統 9を呼出し (ステップ705)、エラーが発生するのを 処理が実行される。エラーが発生した場合の障害情報を 【0058】一方、生成されたサプスレッドでは以下の

重大エラーが発生した場合には装置の運用を停止するた めに、システム停止イベントをメインスレッドに送信し ア晦をアクセスしてチェックする(ステップ706)。 発生したのかを、ファクシミリ通信ボードのハードウェ (ステップ101)、エラー監視スレッドを終了する 【0059】次いで、監視手段119は重大なエラーが (ステップ708)。

を受信バッファから参照し、そのポートのポート制御フ ロセスを再起動することにより行なう。 例えば、障害により制御プロセスが停止したボート番号 5 にジャンプしてエラー監視を統行する。復旧処理は、 復旧処理を行った後(ステップ711)、ステップ70 く、受信したエラーが復旧可能なエラーである場合は、 ップ710)。また、監視停止が実行されたのではな 停止が実行されたのであればスレッドを終了する(ステ 実行されたのかをチェックし(ステップ109)、監視 したエラーが重大エラーでないならば、監視停止手段が 【0060】ステップ706でのチェックの結果、受信

理とシステム停止処理とをメインスレッドで実行する一 ソスフッドセサノスフッド5分割し、通常のイベント処 作であるが、このように監視プロセス106全体をメイ 【0061】以上一連の処理が監視プロセス106の動

> が可能となり、プロセス関通信等を使用して定期的に被 め、エラー検知処理部が非常に簡素化される。 監視プロセスの異常終了を検知をする必要がなくなるた **設ける必要がなくなる。更に、サプスレッドで監視手段** カバリを実行する処理をサプスレッドで実行することに 119を呼び出す(ステップ105)のみでエラー監視 ラー監視をしない処理において、不必要なエラー処理を ない処理とを明確に区別することが可能になるため、エ より、監視プロセス106がエラー監視をする処理とし 方、監視手段119を呼出してエラー検出をしエラーリ

ポート番号を獲得して被監視デバイスファイル(110 実行した後 (ステップ805)、エラーチェックを行い ~112) をオープンする (ステップ802) 。 次い 4)。更に、本来の処理であるファクシミリ通信処理を Sから獲得し(ステップ803)、入出力制御コールを エラー通知フローを示している。まず、制御する通信ポ る。図8は、ポート慰御プロセス(107~109)の セス (107~109) の処理フローについて説明す である通信処理(ステップ805)を繰り返す。 基本OSに発行して登録手段を呼び出す(ステップ80 で、登録すべき情報の一つであるプロセスIDを基本O 一トの初期化処理を行い(ステップ801)、制御する (ステップ806)、エラーがない場合には、通常処理 【0062】次に被監視プロセスであるボート制御プロ

07)、ことによりエラー情報が監視プロセス106に 606)、監視手段を実行状態に設定する(ステップ6 情報が監視プロセスのバッファにコピーされ(ステップ 807)、ポート想御プロセス(107~109)を終 スファイル(110~112)をクローズし(ステップ 【0063】エラーを検知した場合には、被監視デバイ れ、ステップ804で登録手段121に登録しておいた プ807)により図6の使用終了手段122が呼び出さ 『する(ステップ808)。このクローズ処理(ステッ

イルをクローズするので、使用終了手段が確実にコール って監視プロセスが被監視プロセスの実行状況を定期的 情報を監視プロセスに通知するようにしたことにより、 07~109)の動作であり、クローズ処理により障害 費を押さえることができることとなる。 ラー処理部が非常に簡単となり、さらに、計算資源の浪 にプロセス関通信等により監視する必要がなくなり、エ され、監視プロセスに障害情報が確実に通知される。従 例外毎が発生し被監視プロセスが消滅した場合において も、基本OSがオープンされている被監視アパイスファ 【0064】以上一連の処理がポート制御プロセス(1

により、監視プロセスが定期的に被監視プロセスの状態 をチェックすることなく、確実簡易にエラー通知を行う ことが可能となるため、エラー処理部のプログラム量を 【発明の効果】以上の説明から明らかなように、本発明

> プロセス間通信を使用しないので装置の計算資源の浪費 飛躍的に減らすことが可能になり、また、エラー検知に を避けることができる。

が異常終了した場合でも確実にエラー検出を行ない得 的に監視プロセスに通知することにより被監視プロセス 報を被監視プロセスとは異なる場所に保持しておき事後 【0066】更に、具体的には、エラー検出に必要な情 また、システム起動時に際しては強実にエラーチ

ロセス終了時に確実に監視プロセスに通知することが可 後にそのプロセスのエラー監視が可能になる。 【0067】また、エラー通知に必要な情報を被監視プ

ェックの準備が整うことになり、被監視プロセス起動点

セスで生じたエラーを検知することが可能となる。 するため、監視手段を呼出すだけで全ての監視対象プロ 【0068】また、監視手段がエラー検出を開始、統行

要な処理に対してのエラー監視、エラー通知を避けるこ とが可能となる。 し、監視終了手段によりエラー監視を中断でき、監視不 【0069】また、監視手段によりエラー監視を開始

を明確に区別することが可能になるため、エラー監視を しない処理において、不必要なエラー処理を実行する必 【0070】また、エラー監視をする処理としない処理

でも確実にエラー検出、エラー通知を行ない得る。 【0071】また、被監視プロセスが異常終了した場合

イスファイルの利用によりエラー検知を確実に行ない仰 【0072】また、被監視プロセスの状態監視用のデバ

を全て再生できるので、監視プロセスがなく装置の操作 が不能になるという状況をなくすることが可能となる。 【図面の簡単な説明】 【0073】 更に、 リセットにより必要となるプロセス

たFAX情報サービス装置ソフトウエア構成図 【図1】本発明のエラー制御デバイスドライバを使用し

始手段詳細フロー図 【図2】本発明のエラー制御デバイスドライバの使用開

【図3】本発明のエラー制御デバイスドライバの監視手

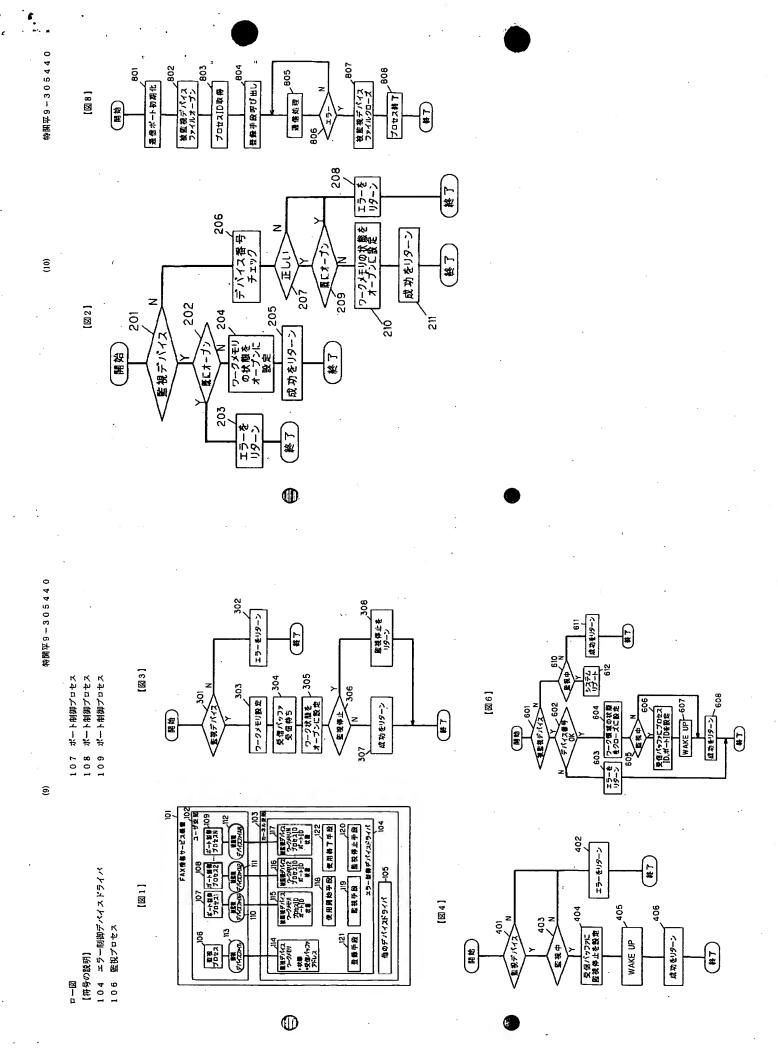
止手段詳細フロー図 【図5】本発明のエラー制御デバイスドライバの登録手 【図4】本発明のエラー制御デバイスドライバの監視体

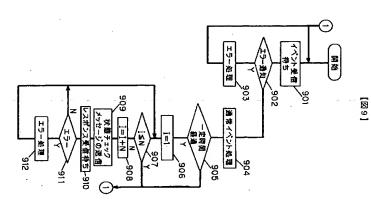
【図6】本発明のエラー制御デバイスドライバの使用終

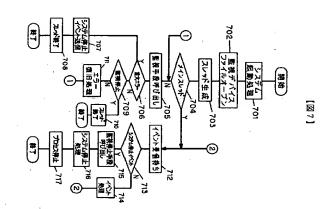
**『手段詳細フロー図** 

【図1】本発明の監視プロセスの処理フロー図 【図8】本発明のポート制御プロセスの処理フロー図 【図9】従来例によるエラー領戌プロセスの詳細フロー

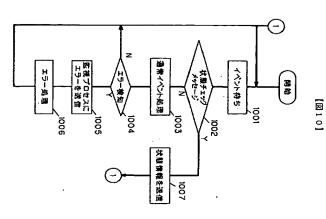
【図10】従来例による被エラー臨視プロセスの詳細フ







1



(12)

(11)

特開平9-305440

特開平9-305440